# Wide Area Information Server Concepts

TMC-202

**Brewster Kahle**
**11/3/89**
**Thinking Machines**

Wide Area Information Servers answer questions over a network feeding information into personal workstations or other servers. As personal workstations become sophisticated computers, much of the role of finding, selecting, and presenting can be done locally to tailor to the users interests and preferences. This paper describes how current technology can be used to open a market of information services that will allow user's workstation to act as librarian and information collection agent from a large number of sources. These ideas form the foundation of a joint project between Apple Computer, Thinking Machines, and Dow Jones. This document is intended for those that are interested in the theoretical concepts and implications of a broad-based information system.

The paper is broken up in three parts corresponding to the three components of the system: the user workstation, the servers, and the protocol that connects them. Whereas a workstation can act as a server, and a server can request information from other servers, it is useful to break up the functionality into client and server roles. A final section in the appendix outlines related systems.

Ideas for this have come from Charlie Bedard, Franklin Davis, Tom Erlickson, Carl Feynman, Danny Hillis, the Seeker group, Jim Salem, Gitta Salomon, Dave Smith, Steve Smith, Craig Stanfill, and others. I am acting as scribe. Comments are welcome (brewster@think.com).

## Table of Contents

# I. Introduction

Distributing knowledge was first done with human memory and oral tradition, later by manuscript, and then by paper books. While paper distribution is still efficient distribution mechanism for some information, electronic transmission makes sense for other. This project attempts to install an electronic "backbone" for distribution of information. Some information is already distributed electronically whether it is printed before it is consumed or not. This project attempts to make electronic networks the distribution technique for more types of information by exploiting new technology and standardizing on an information interchange protocol.

The problems that are being addressed in the design of this system include human interface issues, merging of information of many sources, finding applicable sources of information, and setting up a framework for the rapid proliferation of information servers. Accessing private, group, and public information with one user model implemented on personal workstations is attempted to allow users access to many sources without learning specialized commands. A system for finding information in the sea of possible sources without asking every question of every source can be accomplished by searching descriptions of sources and selecting the sources by hand.

An open protocol for connecting user interfaces on workstations and server computers is critical to the expansion of the available information servers. The success of this system lies in a "critical mass" of users and servers. This protocol, then, could be used on any electronic network from digital networks to phone lines.

For the information owners to make their data available over a server, they must be easy to start, inexpensive to operate, and profitable. One possible approach would be to provide software at a low price that will help those with information holdings to put their data on an electronic network. The power of the current personal workstations is enough to enable sophisticate information servicing capabilities. Charging for services can be done in a number of ways that do not entail setting up large billing operations. In this way, it is easy to set up, operate, and charge for information services.

The key ideas that the WAIS system are that information services should be easily and freely distributed, that the power of the current workstations can provide sophisticated tools as servers and consumers, and that electronic networks should be exploited to distribute information.

## II. The Workstation's Role in WAIS

The personal workstation has grown to be a sophisticated computer that can store hundreds of books worth of information, multiprocess, and communicate over a variety of networks. The advanced capabilities of the workstation are used to find appropriate information for the user by contacting, probing, and negotiating with information servers. The explosion of available information may change the way we use computers since the usual approaches to information on workstations may not grow to make the new information environment understandable. The proposed mechanism involves finding information with one mechanism called "Content Navigation" whether the data is local or remote, available immediately or over time. This section details what a workstation might do to collect and present information from a variety of sources.

### A. Accessing Documents with Content Navigation

Currently, the common way to find a document (or file) is the "Finder" on the Macintosh or most other machines. This tree structure requires the user to remember where s/he has put each file. This approach works when a user is familiar with the file organization. It is also computationally efficient. To aid those that have forgotten the exact location many systems have some way to locate files anywhere in the structure based on the filename ("Find File" on the the Mac, and "find" on Unix machines). The number of potential files increases as the disk space become less expensive and networks let users access remote files. At some point, when the number of files becomes large, this organization can become unwieldy because of the amount the user has to remember.

Another technique that is currently popular is to augment documents with static *HyperText links* [1],[2]. These links help users move through 500 MegaByte CD-ROMs of data without being overwhelmed. HyperText systems allows the author to provide "paths" through the document. The HyperCard system, from Apple, also has a simple content searching mechanism that helps navigate without those links. HyperText links give the author another tool to guide the user and augment the capabilities of the file system.

A different technique that would allow access to a large collection of documents based on *document content and similarity* can be called "Content Navigation." With this tool, documents are retrieved by starting with a question in English. A single line, or headline, would describe possible documents that are appropriate. These documents can be viewed, or used to further direct the search by asking for "more documents like that one". Each document on the disk (or some other source) is then scored on how well it answers the question and the top scoring documents are listed for the user. Since full natural language processing is currently impossible, each document type, be it and newspaper article or a spread sheet, must have some simple measure to determine how relevant it is to the question asked. For text documents a useful and powerful

[1] Nelson, Ted. Literary Machines.

[2] HyperCard by Apple (ref?)

measure is to count the number of words in common between the question and the text. This well known technique of Information Retrieval[1] can be augmented with different weighting schemes for different words or constructions. Other types of information might be retrieved with specific question formats.

Thus, documents can be found by asking the "navigator" for documents that contain a set of words. Those documents that share the most words with the question will come back at the top of the list (have the best "score"). In this system the "answer" to a question is not a single document, rather it is an ordered list of candidate documents.

Content navigation is not new; NeXT and Lotus have implemented systems for personal computers,[2] many text database systems on mini-computers, and the DowQuest system using a super-computer. In general, there is no standardization yet on how these systems should be queried and used.

## B. Dynamic Folders Find Information for the User

Content navigation takes a question and returns an ordered list of possibly relevant documents. The question can be further refined by giving feedback as to how relevant the documents were. The results of a question can be seen as cousin to the file folder in that it contains a list of documents. In reality, the answers to a questions might not be a "copy" of a document, but a "reference" or pointer to a document. These question and answer sessions can be saved just like a file folder can be saved. Saving a session also frees the machine to find answers when the user in not looking. This capability becomes important when some of the questions take time to answer because the data might be far away or difficult to answer. This section discusses one way to think of a saved question: a Dynamic Folder.

"Dynamic Folders" are a cross between a database query and a Macintosh folder that can give us great power in defining questions and probing databases. Text database queries respond with a list of pointers to "hit articles", in the form of titles or headlines, that might interest the user. At that point, the entire article can then be retrieved, if desired. A Dynamic Folder, similarly, has a question that is used to retrieve headlines. Further a Dynamic Folder can be saved and viewed later. Since a folder is a also structure that holds documents so that they can be viewed later, a Dynamic Folder is a folder that has a question associated with it.. In that way a dynamic view acts like a database query in collecting pointers to interesting documents and like a folder in that it can be closed and opened at different times.

A Dynamic Folder's question or "charter" acts as instructions to an active agent as to what what should be put in the folder. This charter gives the folder a mission to keep itself full of appropriate pointers to files or documents. This charter might be as simple as "all files on my personal disk that have a .c suffix", or all mail received in the last day.

In some circumstances, it is important for a Dynamic Folder to contain pointers to a *part of a file* rather than to *an entire file.* Treating parts of files as first class documents is important in systems that group many independent

[1] Salton, Gerald. Introduction to Modern Information Retrieval, McGraw Hill. 1989.

[2] NeXT calls theirs the Digital Librarian, and Lotus calls theirs Megellan (sp?).

documents in one file, such often done with e-mail or news articles. In this way, "documents" and "files" are slightly different.

A Dynamic Folder's contents will change when the charter has changed, at fixed intervals, or when external events happen. The user interface should indicate how current the folder is if it does not always appear up to date. Ideally, when a user changes the charter of a Dynamic Folder, the contents would reflect this instantly. This is possible for local searches and some remote searches. Sometimes, however, changes in the available documents can not be reflected immediately. This is the case when indexing the contents of new files can take a while and is done in the background. Some folders should be updated periodically to reflect new documents in remote databases. For example, a folder that uses the New York Times should be rechecked every day for new articles. Other updates to folders could be done based on events happening such as a new document being stored on the local disk. This could cause all appropriate folders to see if that file is appropriate to add to the contents.

## C. Using Information Servers

Information servers sit on a network and answer questions. A server, whether local or remote, has some database that can be queried and retrieved from. These servers can be easily accessed by a workstation over a network with a standard protocol (see the Protocol section) using the Content Navigation tool to state queries and the Dynamic Folders to hold and coordinate the responses. In this way, a user's sources of information can be seamlessly expanded past the contents of the workstation without an extra conceptual burden on the user. Part of the "charter" of a Dynamic Folder, then, is the servers that it should use. This combination of tools extends the reach of the user while maintaining a consistent view of information. The capabilities of the servers will be discussed more in the server section, but it is important to see at this point that the workstation can be negotiating with a large number of local and remote servers.

## D. Other User Interface Possibilities

The "Dynamic Folder" is just one way to portray the results of a question. Other visual and aural possibilities have been suggested including draw from newspapers, books, library shelves, and sound recordings. This section touches on these possibilities.

Presenting information in *newspaper format* has been tried at the MIT Media Lab (NewsPeek). This approach shows not only a one-line headline, but also the writer, date, place, and first few paragraphs of the article. This format expresses importance by the size of the headline typeface, the organization of the articles on the page, and the amount of text include on the first page. Advertisements also have a place in such a presentation.

Borrowing from *e-mail programs*, listing the possibilities in order of importance has been the technique used by Thinking Machines and NeXT for displaying candidates. Selecting an article brought the text to another window. This interface style allows the user to mark "good" documents to further refine the question. This approach is closely related to the Babyl, Rmail, and Zmail mail handler programs(ref?).

Showing the source of documents *geographically* was suggested by Tom Erikson of Apple. In this approach, a world map can be used to show areas of interest. This might be a good way to initiate browsing if geographical relevance is an important factor to the user. The number of articles concerning or originating from an area can be displayed conveniently.

Presenting documents like *books on a shelf* is a familiar metaphor to librarians. Information about the age of the book, how frequently it has been used, its size, if it is a picture book or monograph or pamphlet, when it was published (by the age of the font) are easily gathered with this presentation. Grabbing a book and looking at it, or looking on the shelves close by are natural reactions in this metaphor. I do not know of any attempts to display information in this way.

Generating a recording of a person reading the top articles can be useful for commuters. With simple skip forward and back capabilities, this might be an effective way to deliver a custom newspaper to someone driving a car. This ideally would be done with a CD player, but a cassette could be used.

The Dynamic Folder is just one possible presentation idea. This area will be an interesting area for research and prototypes.

## E. Advantages of Remote and Local Filtering

When a user subscribes to a remote server, the user can get a complete copy of the database unfiltered, or can instruct the server to filter the documents remotely. Printed newspapers are delivered whole whether all of it is relevant or not. With electronic distribution, one can imagine a user asking for all sports articles but not the business articles. A query is a form of filter that works at the server. A broad query will retrieve a large number of documents that can be further filtered on the personal workstation. The system and protocols can handle filtering at either or both ends.

Local filtering can done by the content navigation on the local disk after the documents have been retrieved. The quality of this filtering will depend on the quality of the content navigator on the local workstation. The filtering might be able to use knowledge about the user that is impractical to deliver to a server. Local filtering gives the user the most flexibility, but it could entail too much communication or too much disk space. How much filtering will be done on the local workstation has tradeoffs that must be made on a server-by-server basis. If the filtering is done locally, then the workstation might have a *subscription* to a server that periodically retrieves the newest articles.

Remote filtering can reduce the communications bandwidth as well as possibly offer better filtering. A server can have better filtering capabilities because it can be database specific as opposed to the workstation's navigator that must be quite general. Remote filtering, just like an interactive query, in initiated by using a question.

As communications, storage, and local computation costs change relative to each other, different filtering structures might make sense.

## F. Local Caching of Documents

Documents that have been retrieved from a server are stored locally on the personal workstation in a *cache*. A cache is a computer architecture term meaning fast, short term storage that helps speed up access by remembering commonly used entries. In this context, a cache would store documents that the user has seen or might want to see so that access to those documents would be faster and easier. A fundamental property of computer caches is that the use of the cache only makes access faster rather than changing any functionality. In certain circumstances, it might be useful to relax this constraint, but this will be seen below. Most interactive queries will only use the cache and local files because the cache will be up-to-date on its information subscriptions. The cache is very important to make queries interactive even though data may have come from remote servers.

The document cache would be stored locally but is shared between all Dynamic Folders. In this way, an article retrieved for one reason could be used in another folder without requiring two copies. A central repository would have to be managed carefully to keep the most relevant articles but not to overload the storage. A quota might be allocated to the cache, and a cache manager would make decisions about what should stay and what should go. Sometimes the user should be consulted, and other times it can be done automatically. The cache manager should keep *header* information on how each document in the cache such as:

(1) what server the document came from,
(2) how big it is,
(3) if it was looked at by the user,
(4) when it was retrieved,
(5) what folders point to it,
(6) if the user asked to keep it permanently,
(7) what the user thought about it ,
(8) how hard is it to retrieve it again,
(9) how to retrieve it again, if at all.

If a document has been deleted from the cache, but it is still being referenced by a Dynamic Folder, the header information should be preserved enough to be able to retrieve the document again. In this way, deleting a document is not a catastrophe.

Since a cache can hold many of the articles seen by a user, the cache is useful in answering retrieving documents based on "I read an article once about..." (In a study of libraries users of scientific journals, about 60% of the articles read were found by browsing, and about 30% were from remembering that they saw it before and they wanted to know more). Supporting this type of question is important for a WAIS interface. The cache can help here by storing all the documents that the user has read. If the cache can not store all of them then it can be instructed as to what type of documents it should keep on hand.

## G. Local Scoring of Competing Servers

Since a Dynamic Folder can get its data from many servers, it must merge this data and present it in a meaningful way to the user. While servers that rate other servers can help determine which server's answers should be valued (see the ***ratings section), these servers only rate the server as a whole and not the individual documents. Furthermore, the article could be very good, just not appropriate to the question. One way to order the responses presented to the user could be based on a "score" that is assigned to each response by the server. Each server might, for instance, judge the appropriateness of its response to the question on a scale of 1-10. These lists from multiple sources could be merged in that order (weighted by the ratings of the servers) and presented to the user. Unfortunately, since a server would want its data to be used, it has every incentive to rate all articles with at 10. Thus, determining how much to trust the server's scores will improve the selection of documents presented to the user.

One possible solution to this problem is to have local scores for servers to augment what the server says. Therefore, if a server always says "this answer is worth 10" and the user never finds it useful, then the personal workstation can lower the trustworthiness of that server's estimation of itself. Saying 10 all the time is the equivalent to crying wolf; if it does it too often, then users will stop listening. In such a scenario, then, all responses from that server could be degraded by 30% before it is used to merge in with the other database's responses. On the other hand, other databases may underrate themselves and should be boosted.

This local scoring can be used to indicate a user's satisfaction with a database and could be used by others to help in rating it. Further, this local score could be used to determine if the server is worth subscribing to or keeping its articles in the cache.

## H. Budgeting the User's Time and Money

Since the users workstation will be spending the users money to contact some servers, a system of accounting and budgeting must be installed so that users get the most value for their money. The trade-offs of time and money can be tricky to try to represent, so a simple system should be attempted first.

The underlying premise is that the computer knows how much it cost to use different services. This can be easy if a service charges for connect time. If a service is reached with a long distance phone call, however this rate could be difficult. (Maybe a server should be set up that knows how much the phone companies charge for different calls.) Further, if a server charges based on the question, there must be a way for the protocol for limiting the amount spent.

Some queries are going to be very important to happen quickly or they are of no use. Working this into the interface can be tricky.

Ideas towards automatic budgeting are still quite primitive. They involve global limits per month, or limits per Dynamic Folder, etc. Should the workstation enforce the limits? Who can override the limits? We need ideas on this one.

# III. The Server's Role in WAIS

Servers sit on networks and answer questions. Successful servers will have some expertise or service that others find useful whether it is primary information, information about other servers, or a service. A file server, a printer, and a human travel agent can all be viewed as forms of servers. This section describes how servers might be used in a Wide Area Information Servers system.

## A. Probing Information Servers

Finding documents (or more generally, information) on one's personal disk is important, but finding relevant information on remote systems would extend the usefulness of personal computers. Currently, most remote database accesses are not integrated with the workstation model using a "glass terminal" interface which does not use the power of the workstation. Some servers look like extensions of the file system and do integrate naturally (such as Sun NFS and AppleShare) but do not provide ways documents based on content. One of the major goals of the WAIS project is to integrate wide area requests in a natural way with local area requests. This section will describe how different information servers could be integrated into this model.

Using the Dynamic Folder, the user creates lasting questions that can collect answers over time from a variety of sources. The charter of a Dynamic Folder includes what sources should be used, which might include the local disk, local special purpose information servers (such as dictionaries etc), AppleShare file servers, and remote databases or WAIS (see the Examples of Information Servers section).

A wide area information server is a computer which provides information on a particular theme to other computers. Servers sit on a network, such as the phone system, the Internet, or X.25, accept connections from other servers or users in order to answer questions in a standard format.

Each information server can be queried at the time the charter is updated, or it can be periodically polled for new information. Newspaper servers, for instance, should be polled to find new articles, while dictionary servers should only be queried once because repeatedly asking the same question is pointless. Thus, the user's workstation keeps information about each server.

While a map, a spread sheet, an airline ticket, or music might be the appropriate reply to a specific query, the initial question is stated in English. A charter (or question) about "Beethoven's choral works" might result in an article from the encyclopedia server, a schedule of concerts from the newspaper server, and recordings from a music server. Depending on the networks used, some responses might be impractical to retrieve, but the architecture allows for any type of information exchange.

A Dynamic Folder can also be used as an information server to other workstations. This simple form of server can enable others to share information easily. This capability should be put into the user interface to encourage people to exchange information. A Dynamic Folder could be "exported" or made available to those that know about it, or "advertised" by adding it to a directory of services. If it is entered into a directory (which is just another information server) then an English description of the folder should be included.

An information server is probed by putting it in the sources section of the folder's charter. These servers can be varied in size, content, and location. Using content navigation and Dynamic Folders we have an metaphor for accessing many types of information servers.

## B. Examples of Information Servers

Information servers, in the broadest sense, answer questions on a particular subject on some network. Electronic networks have been used for years to distribute information in this way. Some of the servers that are available on local area networks have been:

File serving
Printers
Compute servers (such as supercomputers)
FAX
Mail services and archives
Bboard services
Modem pools
Shared databases
Text searching and automatic indexing
CD-ROM servers
Conferencing
Dictionary lookup
User's locations (finger)
Scanners/OCR
35mm Slide output

Wide area networks open up other possibilities for other services. Some services will be offered because they are expensive to offer on a local basis, because it requires some special expertise or machinery, or because it is used infrequently on a local basis. Examples of wide area services that could be offered:

Current newspapers and periodicals
Movie and TV schedules with reviews
Bulletin boards and chat lines
Archive searching through public databases
Hobby specific information (ie sports scores or newletters)
Mail order shopping services
Banking services
Talk services, bboard, and party line styles
Directory information (both online sources and Yellow Pages)
Scientific papers
Government databases, such as patents, congressional record, and laws.
Library catalogs (eg. OCLC)
Weather predictions and maps
Usenet and Arpanet articles
Maps with driving directions included
Software distribution
Remote conferencing
Voice mail
Music and video archives
Pizza ordering

What services will be popular or commercially successful can only be guessed.

## C. Navigating through the "Directory of Services"

The *Directory of Servers* is an information server maintains a database of available servers and how they are contacted. Like the white pages of the phone system the directory should be easy and cheap to use and include everyone. Equally important, this directory is easy to add to. Thus, people with something interesting to offer are encouraged to add their service to the directory.

A *directory entry*, however, should give enough information to understand what the service is and how to connect to it. This entry is similar to a yellow-pages entry in the phone book since the goal is to advertise the service. A directory entry includes:

(1) Description of server in English,
(2) the parent server if it is a subsidiary of a larger server,
(3) related servers,
(4) public encryption key, and
(5) contact information including networks and contact points,
(6) cost information.

A local workstation would keep extra information such as:
(1) locally determined "score" reflecting usefulness
(2) subscription information (if any),
(3) user comments, and
(4) time of last contact.
This information would be used to help determine when and if the server should be contacted, and how the responses should be handled.

Navigating in the sea of servers to find new servers can be done using the content navigation technique. In this way a question on classical music would retrieve documents as well as directory entries. This could be done by storing the directory entries on the local disk (in the cache) and accessing it just like local documents based on the appropriateness of the description. Thus retrieving the document would show all the directory information. In that way, a user that is unaware of a certain server would be presented with a description of that server with a listing of its hits for the current question so that s/he could effectively evaluate its potential value of the server. If the server is added to the list of servers for that viewer, then it would be queried in the future.

Maintaining an up-to-date list of services in the cache naturally falls out of content navigation and Dynamic Folders model because a *directory of services viewer* would have the charter to keep itself up-to-date on directory changes, and can be probed using content navigation. The directory of services viewer would list the remote directory server or servers in the sources slot. That way, the directory is kept locally and is fast to access.

Cost and availability information can help guide the workstation to alert its user to new choices of databases. If a new server appears in the directory that is cheaper than the current server, then it could be suggested as an alternative server. This can be complicated to do well, but the benefits of not having the user cull through new directory listings can warrant work in this direction. As Stewart Brand said, "One of the problems with a market based system is that you are always shopping!" Hopefully, the workstation can do some of the mindless part of comparing servers.

Directories are classically owned and serviced by the communications companies. In this role, the communications company is an unbiased party that profits from the use of the system as a whole. Further, communications companies generally take on a teaching role to get users familiar with the system and aid those with problems. This has been true with AT&T with the telephone, the different phone companies with the 900 numbers, and the Network Information Center for the Arpanet. Whether the communications companies take over this role or not, the directory must be supported by some organization or organizations that profit from the use of the system.

## D. Servers that Rate other Servers

With a large number of servers, it would be nice to know which ones are sponsored by crooks, and which ones are gems. The directory of information servers necessarily accepts all applications for inclusion, just as the white pages do. Unlike the white pages, however, is a description (or advertisement) of the server is included which can be misleading with the result that users are charged for contacting fraudulent servers. Some protection can be offered by independent servers that rate or grade other servers. These servers can serve somewhat the same roles as Consumer Reports, Better Business Bureau, and movie reviewers. This section describes what rating services might do within the WAIS system.

Just as people use movie reviewers to help them select what movies to see, rating services can help in the selection of quality servers. Servers that provide "grades" or reviews of other servers will become useful as the number of servers grow. These ratings can come in many forms such as a numeric grade, formatted reviews that can be used with filters, or a free form discussion. Thresholds can be used by different users to ensure that a server is proven before it is used. This threshold might best be used in conjunction with the cost so that even worthless, but free databases might be tried.

These rating services can come from professional servers or from friends. A user does not have to subscribe to just one rating service, since a combination might be more useful. Combining information from multiple ratings is an interesting topic for exploration.

Creating the ratings server with personal ratings could also be automated somewhat since, each user's workstation keeps track of how frequently a server has been found useful. This information, or any other, can be exported so that other people can select servers that are commonly used.

Numeric ratings of servers can be merged into the user interface by helping order the documents suggested to the user. Therefore, for some user, articles from the Wall Street Journal might get better scores than a similar article in the People's Enquirer. This information could also be displayed by the color of the headline, for instance, so that unrated services would not be overly penalized.

Just as movie goers start to trust a reviewer that has agrees with them on past movies, users will trust rating services that they agree with. Selecting a rating service based on this criteria can have some interesting effects. The rating services that a user has agreed with the most will single themselves out automatically. Users with similar tastes would then find each other. With such an arrangement, one could be lead to find other servers just because other users have liked it whether it is logically related to the common servers or not. This is an automated form of the "if you like this book, then you will like this other book" system. Further, if two users like many of the same things, then they might want to meet.

A generation of server speculators can also arise. Since servers are paid based on people using them, a ratings server will want people to use them often. If agreeing with user's past evaluations is criteria for using a ratings service, then predicting what people will like will be a lucrative business. If a server turns out to be right, then it will be used more. This type of speculation is closely related to the stock market advisers that have become notable of late. A difference would be that this form of speculation is trying to predict what will be interesting to people.

## E. The Role of Editors

One of the conclusions from the NewsPeek personal newspaper project at MIT (I hear) was that editors still had a place in the electronic age by reviewing and selecting certain articles as important. Unlike the rating services, an editor grades specific articles as whether they are important. These grades are similar in many ways to the rating services and might be able to be merged.

A Dynamic Folder might have a charter like: "any article from the front page of the New York Times" which is a command to use what the editor suggests the top articles are. Like the rating services, this can be independent of the sources of the articles and combine the information from multiple sources.

A form of editor server would be if users kept track of their favorite articles and put them in a Dynamic Folder and exported it for others. This way, many favorite servers might emerge and articles could be selected based on friend's suggestions.

Automatically figuring out what the user thought of a document is tricky. Clues as to what the user thought of it are:

(1) how many folders point to it,
(2) if the user read it, how much of it, and for how long,
(3) has the user ever taken any information from it to be used in other documents,
(4) has the user ever referenced it.

This type of information could greatly improve users ability to deal with the flood of available information. Furthermore, throwing away all the thoughts a user has about a document is denying others of that mental effort.

## F. Markets and Hierarchies: Using Silicon Valley

Currently there are several online information providers and many online information "brokers". Brokers provide the connections between the workstations and the information providers (such as PC-link and Compuserve). Sometimes these brokers have services of their own such as electronic mail and bulletin board services. These brokers try provide a complete information environment by providing access to servers. This structure forces a new information server to be connected to many brokers to have their product used since many users only use a few brokers.. The airline reservation program Eaasy Sabre, for example, is available on 20 of these broker networks. The approach of WAIS is to have an open system of interconnection between users and servers where the brokers can act as a server, but is not an all encompassing information environment. With an open system we have a "market" of information servers rather than a controlled environment or a "hierarchy"[1] . Such a structure could open up the field to many more servers and more sophisticated front-ends.

A market based approach would only standardize on the interchange formats leaving different companies free to store and service queries in any way deemed efficient. The user interfaces, similarly, are free to evolve to fit users needs. Since the protocol is not "terminal oriented" (as most systems are today), it frees the computers on either side to be sophisticated in serving the user.

Rapid evolution of a technology can happen in a market system if the structure is designed well. As long as the protocols are flexible enough to start with, and a procedure for changing the protocol is established, then the components will evolve independently by companies seeking to gain a competitive edge.

[1] Malone, Thomas. Electronic Markets Electronic Hierarchies, CACM June 1987 ***Check this.

Silicon valley is an example of a market based system that led to rapid evolution of hardware in the 1970's and software in the 1980's. As the needs of the customers became understood and defined, larger companies that had good marketing and service reputations could make the profitable components without the help of the plethora of small companies. Information servers is an innately niche-based market given the diverse information needs of the population. Furthermore, the industry is more like a service industry than a manufacturing one because of the continual need for updates and new information. For these reasons, the silicon valley structure can help in the rapid evolution of this market.

The key is to have enough users to make the servers profitable. Since, small companies can not wait long before investment turns to profit, achieving early income is important to get the system started. A "critical mass" of users might form if the first interfaces were inexpensive or free, and a few useful servers were available.

## G. How Server Companies Can Make Money

If the WAIS system is to take off, then server companies must be able to make money. Companies that offer servers can make money by billing users directly, using credit cards, or by using 900 numbers to have the phone system bill the users. Direct billing is difficult to set up and can be expensive to operate, but large providers might want to do this. Credit card billing has been a popular one for information providers. This enables any network to connect the user to the server and then the user is charged for use of the server. Typically, the first transaction with a server is a negotiation of how payment will occur and the allocation of a password for future transactions. This could be automated in the WAIS system so that the workstation could know how much the costs will be and keep a total of everything spent. A risk with the credit card system is that a credit card number in the hands of a crook can enable him to make fraudulent charges. With the potentially large number of WAIS systems, this might prove dangerous. Ratings services might be able to help weed out the fraudulent information providers (if any).

Another approach is to use a phone company service over 900 numbers. When a company is assigned one of these numbers, callers are charged per minute of phone conversation and these charges appear on the phone bill every month. Typically the phone company gets 50% of the revenue from this and the charges range from $.10 to $2 per minute (PacBell gets $.25 for the first minute and $.20 thereafter). This approach eliminates the need to have a negotiation of credit card information and limits some of the risks of disclosing a credit card number. On the other hand, the charge for billing is high. Another limitation is that one must use the phone system to connect with the server.

In any case, there is very low overhead in starting a server and earning money. All one needs is a phone, a computer, and some desirable information. This is crucial to the success of the system.

All methods of billing are likely to be used and should be supported by the WAIS interfaces.

# IV. The Protocol's Role in WAIS

> "... they have all one language; and this is only the beginning of what they will do; and nothing that they propose to do will now be impossible for them"
>
> Genesis 11:6

To connect a workstation to a server requires a communication network and a language to talk. The communications network can be anything that allows computers to communicate such as modems, Internet, or digital phone networks. A protocol is the language used to relate questions and receive answers between the workstations and servers. This section describes some of the issues involved in this protocol.

## A. Open Protocols Promotes Wider Acceptance

It is important to the success of this system to have an open protocol that allows users to connect with servers. Several models for how to create an open standard have been tried, such as: have a company own it and license it (Adobe, for instance), have a university develop it (X Windows, for instance), have a standards organization bless it (Common Lisp, for instance), and simply make the specification available and declare is open (IBM PC, for instance). Each approach has advantages and disadvantages. The key point is that certain attributes be adhered to.

1. The companies that are developing the protocol must be open to using existing standards, and not feeling that new protocols should be protected.
2. A system for enhancements to the standard should be set up. Standards committees are often used for this.
3. The standard should be able to transmit data in a variety of formats. There are many emerging multi-media standards. A good standard will be able to transmit these information standards.
4. The query part of the protocol should be able to accept different formats of queries. Queries might, eventually, have multimedia expressions. These should be free to evolve with periodic standardization.
5. The query must have some method to transmit cost restrictions and time-outs. It should also be able to handle query forwarding while avoiding circularities.

An idea for a query language is to use English that is restricted by the constructs that are understood by the servers. As systems become more complicated, they can handle more English constructs. In this way, future server systems can get more information from a query and produce more appropriate responses, simpler systems might use the words in the query without parsing the structure of the query. This approach would allow the servers to change, while the not changing the human interface and the protocols. The English language approach has been very successful for untrained users of the Dow Jones DowQuest system.

The overall success of this system largely depends on how well these protocols work and how they are made available. There is a standard that could solve part of the problem: NISO Z39.50-1988. This standard can help with connecting to servers, delivering queries, and getting responses back. It does not specify the query language or the format of the retrieved records. Other standards may be able to aid other communications needs.

## B. Hardware Independence

Since this system depends on an open protocol rather than a particular implementation, the workstation, servers, and communications systems can all be made up of various hardware technologies that would evolve in time. This independence fosters an appropriate use of all hardware pieces, and a freedom to compete to produce the best components.

Each personal workstation platform has attributes that are appropriate to exploit differently. These can be used to make tailored user interfaces. Further, a competition for the best caching and selection criteria should emerge which will hopefully settle into a good general standard. As personal workstations start to handle audio and video, these can be retrieved with the WAIS system if the bandwidth is available.

Nintendo, for instance, makes a home computer that connects to the television that is installed about 25% of all American homes. They are providing information services to 150,000 Japanese households using this technology. This might be an attractive front-end to a WAIS system.

The server computers will range from personal workstations to supercomputers. Most databases are under 1 gigabyte so they can be stored and processed with a personal workstation unless there are a very large number of users. Supercomputers will be used in applications where there is a large amount of data or there are a very large number of users. Supercomputers can offer superior query handling by doing extensive work on each query.

The communications systems used should be any that are locally available. The bandwidth requirements for text can be satisfied with current phone systems using modems. As advances in bandwidth and connectivity emerge, such as X.25, ISDN, and InterNet; then the range of offerings from the information providers should go up.

Since no component is centralized, this system is free to be established anywhere in the world. Other more centralized systems, such as Minitel, have had difficulty in expanding outside of France. This system should encourage independent regions to set up a compatible system because of the availability of software for servers and workstations.

## C. Protecting the User's Privacy

> "Electrical information devices for universal, tyrannical womb-to-tomb surveillance are causing a very serious dilemma between our claim to privacy and the community's need to know"
> Marshall McLuhan, Media is the Message

To encourage users to trust their personal machines with their data and interests, we must be sure to protect people's sense of privacy. As machines start to learn more about their users and start to contact other machines on their user's behalf, the dangers to privacy are significant. There are technical as well as legal issues involved. This section will cover the technical issues in protecting privacy (any good ref for the legal side?).

There is no easy way to protect a personal workstation if an intruder can get at the keyboard. Since the workstation acts on behalf of the user the potential damage that could be done by a crook at the controls would be worse than is currently possible. Since users will be leaving their computer on all the time so that it can contact servers and be used by other servers, we lose the security of the computer being off at night. One way around this might be to able to turn off input from the user while leaving the computer on to contact servers over the network. If a user knows that she is never around at night or on weekends, then this profile might help lead the system to not trust off hour use and require a password. The assumption so far in personal computers is that the machine stays in a secure physical environment and all protection must be directed to network connections. This is not a safe long term solution, and should be thought through carefully.

Other risks are involved when dealing with networks. There are problems with intruders, spies, and forgers. An intruder will try to read, modify, or destroy data that the user did not intend to leave accessible. Spies will watch the traffic from a user to determine the servers contacted and the content of the messages. A forger will copy password information to act like a different user.

Network intruders can be prevented from reading unwanted data by the user only exporting certain Dynamic Folders to become servers for the outside world. A question is whether we want "group" access as well as "world" access as in the Unix file system or some other layered approach. A Dynamic Folder only contains pointers to information. If the information is on the local disk, should that be accessible by a remote machine? Should those files be protected from being read? If the information came from a remote database, should the requester be required to get it from the source even if a copy is on site? What are the copyright issues here?

Spies can watch communications networks and collect passwords and credit card data if this information is sent in clear text (not encrypted) as well as read the data. A public key system makes sense in this application because the directory information can include a key. Public key systems are those that everyone can lock a message (encrypt) for a recipient, but only the recipient can read it. Presumably the public key system would be used in establishing a connection and a special key for the conversation would be established. Current public key systems are too compute intensive to be used for large volumes of data. A conversation key could be used with DES or some other encryption system that is easier to compute (usrEZ software has a product that runs at 30k characters/second on a MacII). Adoption of such a system early in the WAIS development would ensure that this type of protection is assumed in modern information systems.

Forgers can be foiled with a system of authentication. Authentication is important when the charges are high or when the system is used for ordering goods. One solution is to use a public key signature system that is easy to implement using the public key system (ref the Public Key papers). A signature is passed so that only the sender could have created it.

## V. Conclusion: Why WAIS will Change the World

Historically, when the distribution of information became easier or less expensive, and explosive growth in learning occurred. Wide area information servers are a new way to distribute information. Since anyone with a personal computer, a phone, and some information can be a server, people are free to create and distribute their work in ways that paper distribution made impractical. The current electronic databases, in general, do not have a standard for interchange. Just as the railroads were owned and controlled by relatively few people current database brokers control access and hence the production of data. The highway system was not owned by anyone and the incremental cost to start a new business was very low. Small businesses flourished partly because of this. WAIS systems, similarly, have very low initial costs and low distribution costs which can pave the way to many servers in a short time.

Since the WAIS system is founded on computer to computer communications, new servers that just learn from other servers and produce useful information or analysis can become profitable. Such a server could be thought of as "smart" and the better servers will learn from other servers and from its own mistakes. Thus a distributed "smart" intelligence can be formed.

BBoard systems have not produced any astounding works of literature, I suggest, because it is difficult to reference older works. If older works were easy to find and reference, then people would be more inclined to make better entries. Better entries would get more references and be used more. No BBoard systems, that I know of, make this easy. Since editors, content searching, and archiving are all fundamental parts of the WAIS architecture, we stand a better chance of high quality works being produced.

A large server, or sage, has a role in this distributed system because it can infer correspondences between many pieces of information. Further, large servers will have many users that it can learn from. Users will teach a server what is important just by using the server. Thus a large server will be the place that great new ideas will be created based on lots of existing information. This new form of intelligence, that is formed out of many participating people and machines, is an exciting prospect.

## VI. Related Documents

Blip Culture Hypermedia, Harry Chesley, Apple.

Catalyzing a Market of Wide Area Information Servers, Brewster Kahle.

Wide Area Information Server Demonstration, Brewster Kahle and Charlie Bedard.

Electronic Markets and Electronic Hierarchies, Thomas Malone CACM June 1987.

Introduction to Modern Information Retrieval, Gerald Salton, Cornell. McGraw Hill.

Parallel Free-text search on the Connection Machine, Stanfill and Kahle CACM Dec 1986.

## VII. Appendix: Comparisons to Existing Systems

There are always precedents to any system, this one included. Some are academic and some are commercial; some are computer oriented and some are human services; some are special purpose and some are generally useful.

**A. Compuserve**(of Columbus Ohio, 1-800-848-8199) is a phone based service with about 1000 services with 500,000 PC subscribers. It includes BBoards, hobby services, home shopping, email, multiuser online games, etc. Interestingly, they have contracted with the government to accept Export License Application transactions and other user interface functions. They have "Personal Newspaper" products and deliver data from many publishers. They own a lot of the underlying communication system, but are afraid of ATT and Baby Bells. They are building sophisticated user interfaces for the PCs and MACs.

Compuserve is owned by H&R Block and charges by the minute. They handle their own billing. They have recently bought most of their competitors (The Source, Access, Software House of Cambridge, and Collier-Jackson of Tampa Florida) and are making a fortune. They turned a profit in 4th quarter fiscal 1985 and by the end of fiscal 1986 it recorded a profit of $1.7 million on $100 million revenues and 300,000 users.

Compuserve is the closest model and can be easily accessed with the WAIS system. On the other hand, WAIS helps you find the database you are interested in, does not use a terminal interface (you use your PC with all of its speed), and WAIS offers subscriptions to services where your PC will keep itself informed automatically. Most importantly, WAIS is not "owned" by anyone and is free to grow independently from a centralized company.

(For more technical information I have a book of their services, Thinking Machines has an account, and I have a series of articles describing their business activities.)

**B. Minitel** in France is an outgrowth of the phone company. As an alternative to phone books, users were offered terminals for their homes. Many people took the terminal. By all reports it has been a very popular system. A 1986 news report said: "The directory for Minitel services is now the size of a phone directory for a small city, evidence that Minitel is a success." George Nahon, managing directory of Intelmatique: "Then need to create a market of users emerged as a prerequisite for a service." One reports speculated that France has put about $500 million into the system by 1986.

Their interface is a terminal type interface and the servers are both human and machine. [Europe is the most exciting continent for information services. It seems that they take this very seriously, while the US government has yet to take the bold steps of investment and standardization.]

**C. NetLib** is a free Unix utility for distributing files through the email. Anyone that has access to the servers via electronic mail can make inquiries and file requests. This system currently has about 100 (a guess) collections world-wide and is growing. In 1987, about 10,000 requests per month were serviced. The bulk of the offerings are software programs rather than raw data. Since no charges are made for queries or requests this system is used by academics and researchers. ATT and Argonne labs are supporting this work.

The automatic reply system (remote-machine-to-local-machine rather than remote-machine-to-local-human interface) in NetLib is similar to the WAIS system. WAIS, however, is not centered solely around EMail as a transport layer; it uses the phone system as well for interactive use. Also, WAIS would help find databases that are relevant and handle the queries and requests through a more "user friendly" interface. (For more on NetLib see Distribution of Mathematical Software via Electronic Mail in Communications of the ACM May 1987)

**D. Switzerland system** Still assessing this system.

**E. Lotus and NeXT text system**

Both Lotus and NeXT have text searching systems that are similar to Thinking Machine's Dow Jones system, but are based on local data (LAN based). Since disks hold close to 1 gigabyte these days, and the entire CM at Dow Jones holds 1 gigabyte, we are close in scope but not performance. On the other hand, a PC will serve its 20 users adequately and the new daily information can be effectively distributed from Dow Jones and other places. Lotus seems to be getting into the information distribution business and is writing software to process that data locally.

These companies see themselves as critically involved in this area. I believe cooperating with them is in our best interest.

**F. Information Brokers**

Many companies act as brokers to other information providers. Often these services will offer electronic mail and bulletin boards. These private systems rarely communicate with each other. The systems that I know of are listed below. If anyone has any information on these or other companies, please tell me.

| | | |
|---|---|---|
| AppleLink(Personal Edition) | 1-800-227-6364 | getting info |
| Delphi | 1-800-544-4005 | getting info |
| Dialcom, Inc. | 1-800-435-7342 | |
| GE Information Services | 1-800-433-3683 | getting info |

This company services the fortune 500 companies with network and processing services using Honeywell and IBM mainframes. They lease lines from ATT and provide an environment for their customers including network services and value added filtering and massaging of data.

| | | |
|---|---|---|
| GEnie | 1-800-638-9636 | getting info |
| IBM Information Network | 1-800-IBM-2468 ext 100 | |
| INet 2000/TravelNet | 1-800-267-8480 | bad number |
| Inet | 1-800-322-INET | |
| NWI | 1-800-624-5916 | |

Quantum Computer Services since 1985, privately held, "multimillion dollars" official commodore info service. Has been supported by commodore.

| | | |
|---|---|---|
| PC-link | 1-800-458-8532 | IBM PC product |
| Q-Link | 1-800-392-8200 | Commodore product |
| America online | | Mac product |

| | |
|---|---|
| Snet | 1-800-272-SNET Dept AA |
| The Source | 1-800-336-3366 |
| StarText | 1-817-390-7905 |
| Travel+Plus | 1-800-544-4005 |
| US videotel | 1-713-323-3000 |
| Western Union EasyLink | 1-800-779-1111 Dept 31 |
| Minitel Services | 1-914-694-6266 |
| Omnet/SCIENCEnet | 1-617-265-9230 |

Other systems that I would like to find out more about:

Holland system, Prodigy, Knight Ridder, Audio Tex, Airline Reservations system, Hospital Ordering System, Verity, Personal Newspaper (Media lab), Information Lens (Media Lab), SuperText.

**G. Hypertext**

Hypertext and WAIS share many attributes for accessing textual information. In some sense, WAIS is an attempt at a large-scale hypertext system by allowing links to be deduced at run-time and across many databases stored in many places. Since servers provide pointers to documents, a pointer to a document can be put in a document and retrieved at a later time. Thus document pointers can be thought of as a crude form of hypertext link.

This form of deducing hypertext links through content navigation might lead to interesting paths that are tailored to a particular user. Automatic systems will never replace the value of having users suggesting links. Suggested links can be added directly to the documents (as in most hypertext systems) or then can be made available in a distributed manner through the favorites databases. In this way, users that found certain articles to be similar or usefully viewed together can put them in a folder and export it as a database. One might ask, "Does anyone have these documents grouped in a server, and if so, what other documents are in that server?" These databases could then be used by others as evidence that they belong together. By combining many people's groupings, one can navigate through large number of documents in potentially interesting ways in a hypertext style.